



SOAP-over-UDP Version 1.1

Committee Specification 01

14 May 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/cs-01/wsdd-soapoverudp-1.1-spec-cs-01.html>
<http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/cs-01/wsdd-soapoverudp-1.1-spec-cs-01.docx>
(Authoritative Format)
<http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/cs-01/wsdd-soapoverudp-1.1-spec-cs-01.pdf>

Previous Version:

<http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/cd-03/wsdd-soapoverudp-1.1-spec-cd-03.html>
<http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/cd-03/wsdd-soapoverudp-1.1-spec-cd-03.docx>
<http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/cd-03/wsdd-soapoverudp-1.1-spec-cd-03.pdf>

Latest Version:

<http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/wsdd-soapoverudp-1.1-spec.html>
<http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/wsdd-soapoverudp-1.1-spec.docx>
<http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/wsdd-soapoverudp-1.1-spec.pdf>

Technical Committee:

OASIS Web Services Discovery and Web Services Devices Profile (WS-DD) TC

Chair(s):

Toby Nixon, Microsoft Corporation
Alain Regnier, Ricoh Company Limited

Editor(s):

Ram Jeyaraman, Microsoft Corporation

Declared XML Namespace(s):

None.

Abstract:

This specification defines a binding for SOAP envelopes to use datagrams.

Status:

This document was last revised or approved by the WS-DD TC on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/ws-dd/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/ws-dd/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/ws-dd/>.

Notices

Copyright © OASIS® 2009. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction	4
1.1	Requirements	4
1.2	Terminology	4
1.2.1	Notational Conventions	4
1.2.2	Terms and Definitions	5
1.3	XML Namespaces	5
1.4	Relationship to Web Service Specifications	5
1.5	Normative References	5
2	UDP Packet	7
2.1	Source Address and Port	7
2.2	Data Octets	7
3	Message Patterns	8
3.1	One-way	8
3.1.1	One-way Example	8
3.2	Request-response	8
3.2.1	Anonymous [reply endpoint]	8
3.2.2	Request Example 1	9
3.2.3	Response Example 1	9
3.2.4	Request Example 2	9
3.2.5	Response Example 2	10
3.3	Multicast	10
3.4	Retransmission	11
4	Message Encoding	12
5	URI Scheme	13
5.1	Syntax	13
5.2	Semantics	13
6	Security Considerations	14
7	Conformance	15
Appendix A.	Example retransmission algorithm (non-normative)	16
Appendix B.	Example duplicate detection mechanisms (non-normative)	17
Appendix C.	Acknowledgements	18
Appendix D.	Revision History	20

1 Introduction

Many application protocol patterns match the semantics of the User Datagram Protocol (UDP) [RFC 768]. Some do not require the delivery guarantees of TCP while others make use of multicast transmission. In order to allow Web services to support these patterns, we need a way to map SOAP envelopes to user datagrams. This support is essential for services using WS-Discovery, where the use of multicast and need for low connection overhead makes UDP a natural choice. It is anticipated that other protocols will have similar requirements. This specification defines a binding of SOAP to user datagrams, including message patterns, addressing requirements, and security considerations.

1.1 Requirements

This specification intends to meet the following requirements:

- Support a one-way message-exchange pattern (MEP) where a SOAP envelope is carried in a user datagram.
- Support a request-response message-exchange pattern (MEP) where SOAP envelopes are carried in user datagrams.
- Support multicast transmission of SOAP envelopes carried in user datagrams.

Support both SOAP 1.1 [SOAP 1.1] and SOAP 1.2 [SOAP 1.2 Part 1] Envelopes.

1.2 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC 2119].

1.2.1 Notational Conventions

This specification uses the following syntax to define normative outlines for messages:

The syntax appears as an XML instance, but values in italics indicate data types instead of literal values.

Characters are appended to elements and attributes to indicate cardinality:

- “?” (0 or 1)
- “*” (0 or more)
- “+” (1 or more)
- The character “|” is used to indicate a choice between alternatives.
- The characters “[” and “]” are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- Ellipses (i.e., “...”) indicate points of extensibility. Additional children and/or attributes MAY be added at the indicated extension points but MUST NOT contradict the semantics of the parent and/or owner, respectively. If a receiver does not recognize an extension, the receiver SHOULD ignore the extension.
- XML namespace prefixes (see Table 1) are used to indicate the namespace of the element being defined.

Elsewhere in this specification, the characters “[” and “]” are used to call out references and property names. This specification uses the **[action]** and Fault properties [WS-Addressing] to define faults.

39 1.2.2 Terms and Definitions

40 Defined below are the basic definitions for the terms used in this specification.

41 Receiver

42 The endpoint terminating a SOAP/UDP datagram

43 Sender

44 The endpoint originating a SOAP/UDP datagram

45 SOAP/UDP datagram

46 A user datagram containing a SOAP envelope in the data octets

47 User datagram

48 A User Datagram Protocol (UDP) packet

49 1.3 XML Namespaces

50 The following lists XML namespaces that are used in this specification. The choice of any namespace
51 prefix is arbitrary and not semantically significant.

52 **Table 1. Prefix and XML namespaces used in this specification**

Prefix	XML Namespace	Specification(s)
s	(Either SOAP 1.1 or 1.2)	(Either SOAP 1.1 or 1.2)
s11	http://schemas.xmlsoap.org/soap/envelope/	[SOAP 1.1]
s12	http://www.w3.org/2003/05/soap-envelope	[SOAP 1.2 Part 1]
a	http://www.w3.org/2005/08/addressing	[WS-Addressing]

53 1.4 Relationship to Web Service Specifications

54 This specification provides a binding appropriate for:

- 55 • SOAP 1.1 [SOAP 1.1]
- 56 • SOAP 1.2 [SOAP 1.2 Part 1]

57 Messages conforming to either SOAP specification can use this binding. This specification relies on WS-
58 Addressing [WS-Addressing].

59 1.5 Normative References

60 [RFC 768]

61 J. Postel, "User Datagram Protocol," RFC 768, <http://www.ietf.org/rfc/rfc768.txt>, August 1980.

62 [RFC 2119]

63 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119,
64 <http://www.ietf.org/rfc/rfc2119.txt>, March 1997.

65 [RFC 2365]

66 D. Meyer, "Administratively Scoped IP Multicast," RFC 2365, <http://www.ietf.org/rfc/rfc2365.txt>,
67 July 1998.

68 [RFC 3986]

69 T. Berners-Lee, et al, "Uniform Resource Identifiers (URI): Generic Syntax", IETF RFC 3986,
70 <http://www.ietf.org/rfc/rfc3986.txt>, January 2005.

- 71 **[RFC 791]**
72 "Internet Protocol (IPv4)", IETF [RFC 791](http://www.ietf.org/rfc/rfc791.txt), <http://www.ietf.org/rfc/rfc791.txt>, September 1981.
- 73 **[RFC 2460]**
74 S. Deering, et al, "Internet Protocol, Version 6 (IPv6) Specification", IETF [RFC 2460](http://www.ietf.org/rfc/rfc2460.txt),
75 <http://www.ietf.org/rfc/rfc2460.txt>, December 1998.
- 76 **[SOAP 1.1]**
77 W3C Note, "[Simple Object Access Protocol \(SOAP\) 1.1](http://www.w3.org/TR/2000/NOTE-SOAP-20000508)", [http://www.w3.org/TR/2000/NOTE-](http://www.w3.org/TR/2000/NOTE-SOAP-20000508)
78 [SOAP-20000508](http://www.w3.org/TR/2000/NOTE-SOAP-20000508), 08 May 2000.
- 79 **[SOAP 1.2 Part 1]**
80 W3C Recommendation, "[SOAP Version 1.2 Part 1: Messaging Framework \(Second Edition\)](http://www.w3.org/TR/2007/REC-soap12-part1-20070427)",
81 <http://www.w3.org/TR/2007/REC-soap12-part1-20070427>, April 2007.
- 82 **[WS-Addressing]**
83 W3C Recommendation, "[Web Services Addressing 1.0 - Core](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509)", [http://www.w3.org/TR/2006/REC-](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509)
84 [ws-addr-core-20060509](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509), 9 May 2006.
- 85 **[WS-Security]**
86 OASIS Standard, "[Web Services Security Core specification 1.1](http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf)", [http://www.oasis-](http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf)
87 [open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf](http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf),
88 February 2006.
- 89 **[XML 1.0]**
90 W3C Recommendation, "[Extensible Markup Language \(XML\) 1.0 \(Fourth Edition\)](http://www.w3.org/TR/2006/REC-xml-20060816)",
91 <http://www.w3.org/TR/2006/REC-xml-20060816>, 16 August 2006.

92 2 UDP Packet

93 Except as noted explicitly below, this specification does not constrain RFC 768 [RFC 768].

94 2.1 Source Address and Port

95 For security reasons, the source address MUST be supplied at the IP packet level and MUST be the IPv4
96 [RFC 791] address (including but not limited to unicast, multicast, and broadcast addresses) or IPv6 [RFC
97 2460] address (including but not limited to unicast and multicast addresses) of the sender; the receiver
98 SHOULD reject IP packets containing a SOAP/UDP datagram that have inappropriate values for the
99 source address.

100 2.2 Data Octets

101 The data octets MUST contain a SOAP envelope [SOAP 1.1] [SOAP 1.2 Part 1]. The SOAP envelope
102 MUST fit within a single datagram, that is, it MUST be small enough that the overall datagram is less than
103 65,536 (2^{16}) octets.

104 The SOAP envelope MUST use the mechanisms defined in WS-Addressing [WS-Addressing].

3 Message Patterns

106 This specification supports the following message patterns:

- 107 • Unicast one-way
- 108 • Multicast one-way
- 109 • Unicast request, unicast response
- 110 • Multicast request, unicast response

111 as detailed in the rest of this section.

112 This specification uses the constructs **[action]**, **[destination]**, **[message id]**, **[reply endpoint]**,
113 **[address]** in WS-Addressing [\[WS-Addressing\]](#). SOAP messages transmitted over UDP MUST have a
114 **[message id]** property.

3.1 One-way

116 The one-way message is sent in a user datagram.

3.1.1 One-way Example

```
118 (01) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"  
119 (02)           xmlns:a="http://www.w3.org/2005/08/addressing">  
120 (03)   <s:Header>  
121 (04)     <a:To>http://fabrikam.com/Server</a:To>  
122 (05)     <a:Action>http://fabrikam.com/Probe</a:Action>  
123 (06)     <a:MessageID>  
124 (07)       urn:uuid:1da72f1a-5546-493c-934c-a9e3577e206a  
125 (08)     </a:MessageID>  
126 (09)   </s:Header>  
127 (10)   <s:Body>  
128 (11)     ...  
129 (12)   </s:Body>  
130 (13) </s:Envelope>
```

131 This example shows a one-way SOAP message. Lines 01-03 are standard SOAP elements. Lines 04-08
132 specify various WS-Addressing headers. Note that despite the fact that the **[destination]** for the message
133 is specified using a URI that uses the http scheme, the message is still transmitted over UDP. Lines 09-13
134 show standard SOAP elements.

3.2 Request-response

136 The request message is sent in one user datagram and the corresponding response message is sent in
137 another user datagram.

3.2.1 Anonymous [reply endpoint]

139 WS-Addressing defines a URI, "http://www.w3.org/2005/08/addressing/anonymous", that can
140 appear in the **[address]** property of an endpoint reference. If the **[reply endpoint]** property of a SOAP
141 message transmitted over UDP has an **[address]** property with this value, the UDP source address (and
142 source port) is considered to be the address to which reply messages should be sent.

143 The implied value of the **[reply endpoint]** property for SOAP messages transmitted over UDP is an
144 endpoint reference with an **[address]** property whose value is
145 "http://www.w3.org/2005/08/addressing/anonymous".

146 3.2.2 Request Example 1

```
147 (01) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"  
148 (02)           xmlns:a="http://www.w3.org/2005/08/addressing">  
149 (03) <s:Header>  
150 (04)   <a:To>http://fabrikam.com/Server</a:To>  
151 (05)   <a:Action>http://fabrikam.com/Probe</a:Action>  
152 (06)   <a:MessageID>  
153 (07)     urn:uuid:9ceada16-2403-4404-a8cc-60799acd9d1c  
154 (08)   </a:MessageID>  
155 (09)   <a:ReplyTo>  
156 (10)     <a:Address>  
157 (11)       http://www.w3.org/2005/08/addressing/anonymous  
158 (12)     </a:Address>  
159 (13)   </a:ReplyTo>  
160 (14) </s:Header>  
161 (15) <s:Body>  
162 (16)   ...  
163 (17) </s:Body>  
164 (18) </s:Envelope>
```

165 This example shows a request SOAP message. Lines 01-03 are standard SOAP elements. Lines 04-13
166 specify various WS-Addressing headers. Note that despite the fact that the **[destination]** for the message
167 is specified using a URI that uses the http scheme, the message is still transmitted over UDP. Line 09
168 shows a **[reply endpoint]** header specifying the anonymous URI (see Section 3.2.1). Lines 14-18 show
169 standard SOAP elements.

170 3.2.3 Response Example 1

```
171 (01) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"  
172 (02)           xmlns:a="http://www.w3.org/2005/08/addressing">  
173 (03) <s:Header>  
174 (04)   <a:To>  
175 (05)     http://www.w3.org/2005/08/addressing/anonymous  
176 (06)   </a:To>  
177 (07)   <a:Action>http://fabrikam.com/ProbeMatch</a:Action>  
178 (08)   <a:MessageID>  
179 (09)     urn:uuid:5a6ed11a-7a80-409a-82bf-43c4c5092911  
180 (10)   </a:MessageID>  
181 (11)   <a:RelatesTo>  
182 (12)     urn:uuid:9ceada16-2403-4404-a8cc-60799acd9d1c  
183 (13)   </a:RelatesTo>  
184 (14) </s:Header>  
185 (15) <s:Body>  
186 (16)   ...  
187 (17) </s:Body>  
188 (18) </s:Envelope>
```

189 This example shows a response SOAP message. Lines 01-03 are standard SOAP elements. Lines 04-13
190 specify various WS-Addressing headers. Note that the **[destination]** for the message is specified as the
191 anonymous URI. Line 11 shows a **[relationship]** header indicating that this message is a reply to the
192 example message in Section 3.2.2. Lines 14-18 show standard SOAP elements.

193 3.2.4 Request Example 2

```
194 (01) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"  
195 (02)           xmlns:a="http://www.w3.org/2005/08/addressing" >  
196 (03) <s:Header>  
197 (04)   <a:To>soap.udp://fabrikam1.com:54321/Server</a:To>  
198 (05)   <a:Action>http://fabrikam1.com/Probe</a:Action>  
199 (06)   <a:MessageID>  
200 (07)     urn:uuid:9ceada16-2403-4404-a8cc-60799acd9d1c
```

```

201 (08) </a:MessageID>
202 (09) <a:ReplyTo>
203 (10) <a:Address>
204 (11) soap.udp://fabrikam2.com:54322/Client
205 (12) </a:Address>
206 (13) </a:ReplyTo>
207 (14) </s:Header>
208 (15) <s:Body>
209 (16) ...
210 (17) </s:Body>
211 (18) </s:Envelope>

```

212 This example shows a request SOAP message. Lines 01-03 are standard SOAP elements. Lines 04-13
213 specify various WS-Addressing headers. Note that the **[destination]** for the message is specified using a
214 URI that uses the soap.udp scheme. Line 09 shows a **[reply endpoint]** header containing an
215 addressable URI that uses the soap.udp scheme. Lines 14-18 show standard SOAP elements.

216 3.2.5 Response Example 2

```

217 (01) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
218 (02) xmlns:a="http://www.w3.org/2005/08/addressing">
219 (03) <s:Header>
220 (04) <a:To>
221 (05) soap.udp://fabrikam2.com:54322/Client
222 (06) </a:To>
223 (07) <a:Action>http://fabrikam.com/ProbeMatch</a:Action>
224 (08) <a:MessageID>
225 (09) urn:uuid:5a6ed11a-7a80-409a-82bf-43c4c5092911
226 (10) </a:MessageID>
227 (11) <a:RelatesTo>
228 (12) urn:uuid:9ceada16-2403-4404-a8cc-60799acd9d1c
229 (13) </a:RelatesTo>
230 (14) </s:Header>
231 (15) <s:Body>
232 (16) ...
233 (17) </s:Body>
234 (18) </s:Envelope>

```

235 This example shows a response SOAP message. Lines 01-03 are standard SOAP elements. Lines 04-13
236 specify various WS-Addressing headers. Note that the **[destination]** for the message contains an
237 addressable URI that uses the soap.udp scheme. Line 11 shows a **[relationship]** header indicating that
238 this message is a reply to the example message in Section 3.2.4. Lines 14-18 show standard SOAP
239 elements.

240 3.3 Multicast

241 The message patterns defined above can be used with unicast or multicast transmission of UDP
242 datagrams with the following restriction: The response in a request-response message pattern MUST
243 NOT be multicast.

244 Note that in the case of a multicast request, unicast response MEP, the sender of the request might
245 receive multiple responses.

246 Multicast SOAP/UDP datagrams SHOULD be scoped to ensure they are not forwarded beyond the
247 boundaries of the administrative system. For IPv4, this can be done with either time-to-live (TTL) field or
248 administrative scopes [\[RFC 2365\]](#) depending on what is implemented in the network. For IPv6, this can
249 be done by setting the hop-limit field. If either IPv4 TTL or IPv6 hop-limit is used, it is RECOMMENDED
250 that its value be set to 1.

251 The destination IP address of a multicast message MUST be a multicast group.

252 **3.4 Retransmission**

253 To avoid repeated packet collisions, any retransmission implementation SHOULD observe good practices
254 such as using exponential back-off algorithms and spreading. An implementation MAY use the algorithm
255 defined in Appendix A. For each transmission of such a message, the value of the **[message id]** property
256 MUST be the same.

257 **4 Message Encoding**

258 The algorithm defined in Appendix F of XML 1.0 [[XML 1.0](#)] SHOULD be used to determine whether a
259 message is encoded as XML. If use of said algorithm does not result in an XML serialization, the
260 encoding is undefined.

261 5 URI Scheme

262 This section defines a URI scheme for UDP endpoints. The scheme allows hostname and port to be
263 specified. Resolving such a URI provides the information needed to send messages to a UDP endpoint
264 per the protocol defined in this document.

265 5.1 Syntax

266 The syntax of the URI scheme is as follows:

```
267 soap.udp://<host>:<port>[/<rel_path>][?<query>]
```

268 The syntax and interpretation of the host, port, rel_path and query portions is as defined in RFC 3986
269 [[RFC 3986](#)].

270 5.2 Semantics

271 The semantics of resolving a soap.udp URI are as follows:

- 272 1. Use the port portion as the port number.
- 273 2. Resolve the host portion to an IP address.
- 274 3. Using the message protocol defined in this document, send a message to the IP address determined
275 in step 2 using the port number determined in step 1.

276 6 Security Considerations

277 It is recommended that all messages be secured using the mechanisms described in [\[WS-Security\]](#) to
278 prevent tampering or falsification.

279 All critical headers, such as those described in [\[WS-Addressing\]](#), and the message body, need to be
280 included in signatures to bind all parts of the message together.

281 Recipients SHOULD verify that the sender has the right to speak for the specified source or response
282 location (if one is provided).

283 Messages SHOULD be accepted and processed only from trusted sources (either directly trusted or
284 indirectly trusted via third parties).

285 The UDP packet size introduces a challenge for secure messages due to its limited size. For this reason
286 it is recommended that security tokens not be passed but referenced using the Key Identifier mechanisms
287 described in [\[WS-Security\]](#).

288 SOAP messages containing a **[reply endpoint]** property transmitted over UDP MAY be rejected by a
289 recipient due to security concerns such as distributed denial-of-service attacks.

290 **7 Conformance**

291 A conformant implementation **MUST** satisfy all the **MUST** or **REQUIRED** level requirements defined
292 herein.

293 **Appendix A. Example retransmission algorithm (non-**
294 **normative)**

295 Constants referenced within the algorithm are defined in Table 2 (for unicast messages) and Table 3 (for
296 unreliable multicast messages). The value of those constants (as defined in Table 2 and Table 3) is non-
297 normative. Implementations and other specifications MAY override the value of those constants.

298 **Retry and back-off algorithm.**

- 299 1. Transmit;
300 2. if *_UDP_REPEAT <= 0 go to Step 11;
301 3. else *_UDP_REPEAT--;
302 4. Generate a random number T in [UDP_MIN_DELAY .. UDP_MAX_DELAY];
303 5. Wait T milliseconds;
304 6. Retransmit;
305 7. if *_UDP_REPEAT <= 0 goto Step 11;
306 8. else *_UDP_REPEAT--;
307 9. $T = T * 2$; If $T > \text{UDP_UPPER_DELAY}$ then $T = \text{UDP_UPPER_DELAY}$;
308 10. go to 5;
309 11. Done.

310 **Table 2. Protocol retry and back-off constants for unicast messages**

Constant / Message	Value
UNICAST_UDP_REPEAT	1
UDP_MIN_DELAY	50
UDP_MAX_DELAY	250
UDP_UPPER_DELAY	500

311 **Table 3. Protocol retry and back-off constants for unreliable multicast messages**

Constant / Message	Value
MULTICAST_UDP_REPEAT	2
UDP_MIN_DELAY	50
UDP_MAX_DELAY	250
UDP_UPPER_DELAY	500

312 **Appendix B. Example duplicate detection**
313 **mechanisms (non-normative)**

314 A receiver keeps a list of the last n messages received along with the **[message id]** property [[WS-Addressing](#)]
315 associated with each message. When a new (non-duplicate) message arrives, the oldest
316 message is removed from the list.

317 A receiver tracks all messages received in the last x milliseconds along with the **[message id]** property
318 [[WS-Addressing](#)] associated with each message. Messages received more than x milliseconds ago are
319 removed from the list.

320 For both approaches any message arriving with a **[message id]** property identical to one of those the
321 receiver has in its list is a duplicate. Messages with unique values for the **[message id]** property are not
322 duplicates.

323 The timestamp specified in the Security header block [[WS-Security](#)] MAY be used to limit the duration for
324 which **[message id]** properties need to be remembered.

325

Appendix C. Acknowledgements

326 The following individuals have participated in the creation of this specification and are gratefully
327 acknowledged:

328 **Participants:**

329 Geoff Bullen, Microsoft Corporation
330 Steve Carter, Novell
331 Dan Conti, Microsoft Corporation
332 Doug Davis, IBM
333 Scott deDeugd, IBM
334 Oliver Dohndorf, Technische Universitat Dortmund
335 Dan Driscoll, Microsoft Corporation
336 Colleen Evans, Microsoft Corporation
337 Max Feingold, Microsoft Corporation
338 Travis Grigsby, IBM
339 Francois Jammes, Schneider Electric
340 Ram Jeyaraman, Microsoft Corporation
341 Mike Kaiser, IBM
342 Supun Kamburugamuva, WSO2
343 Devon Kemp, Canon Inc.
344 Akira Kishida, Canon Inc.
345 Jan Krueger, Technische Universitaet Dortmund
346 Mark Little, Red Hat
347 Dr. Ingo Lueck, Technische Universitaet Dortmund
348 Jonathan Marsh, WSO2
349 Carl Mattocks
350 Antoine Mensch
351 Jaime Meritt, Progress Software
352 Vipul Modi, Microsoft Corporation
353 Anthony Nadalin, IBM
354 Tadahiro Nakamura, Canon Inc.
355 Masahiro Nishio, Canon Inc.
356 Toby Nixon, Microsoft Corporation
357 Shin Ohtake, Fuji Xerox Co., Ltd.
358 Venkat Reddy, CA
359 Alain Regnier, Ricoh Company, Ltd.
360 Hitoshi Sekine, Ricoh Company, Ltd.
361 Yasuji Takeuchi, Konica Minolta Business Technologies
362 Hiroshi Tamura, Ricoh Company, Ltd.
363 Minoru Torii, Canon Inc.
364 Asir S Vedamuthu, Microsoft Corporation
365 David Whitehead, Lexmark International Inc.
366 Don Wright, Lexmark International Inc.
367 Prasad Yendluri, Software AG, Inc.
368 Elmar Zeeb, University of Rostock
369 Gottfried Zimmermann

370

371 **Co-authors of the initial contributions:**

372 This document is based on initial contributions to the OASIS WS-DD Technical Committee by the
373 following co-authors.

374 Harold Combs, Lexmark International Inc.
375 Martin Gudgin (editor), Microsoft Corporation
376 John Justice, Microsoft Corporation

377 Gopal Kakivaya, Microsoft Corporation
378 David Lindsey, Lexmark International Inc.
379 David Orchard, BEA Systems, Inc.
380 Alain Regnier, Ricoh Company Limited
381 Jeffrey Schlimmer, Microsoft Corporation
382 Stacy Simpson, Lexmark International Inc.
383 Hiroshi Tamura, Ricoh Company Limited
384 Don Wright, Lexmark International Inc.
385 Kenny Wolf, Microsoft Corporation
386

387 **Acknowledgements of the initial contributions:**

388 This specification has been developed as a result of joint work with many individuals and teams,
389 including:

390
391 Erik Christensen, Microsoft Corporation
392 David Langworthy, Microsoft Corporation
393 Yaniv Pessach, Microsoft Corporation
394 Stefan Pharies, Microsoft Corporation
395 Sam Rhodus, Lexmark International Inc.
396 Jerry Thrasher, Lexmark International Inc.
397 Mike Vernal, Microsoft Corporation
398 Elliot Waingold, Microsoft Corporation
399 Dave Whitehead, Lexmark International Inc.

400

Appendix D. Revision History

401

[optional; should not be included in OASIS Standards]

402

Revision	Date	Editor	Changes Made
wd-01	September 16, 2008	Ram Jeyaraman	Created the initial working draft by converting the input specification to OASIS template.
wd-02	September 29, 2008	Ram Jeyaraman	Updated document identifier, added co-chair and editor names, use of urn:uuid (issue 50) in examples.
wd-03	January 15, 2009	Ram Jeyaraman	Created working draft 03 by applying the proposed resolutions of the following issues to CD-01 version: 116 - Update references and related changes 136 - SOAP-over-UDP - UNICAST_UDP_REPEAT and MULTICAST_UDP_REPEAT constant values Updated copyright year to 2009. Appendix C (Acknowledgements). Included a list of TC participants.
wd-04	February 09, 2009	Ram Jeyaraman	Editorial corrections: Converted citations to hyperlinks, fixed links to tables, fixed Appendix heading.
wd-05	April 09, 2009	Ram Jeyaraman	Updates to Appendix C (Acknowledgements). Changes resulting from issue pr007.

403